# Matching Pursuit-Based Compressive Sensing in a Wearable Biomedical Accelerometer Fall Diagnosis Device

Gibson, Ryan; Amira, Abbes; Ramzan, Naeem; Casaseca, Juan Pablo; Pervez, Zeeshan

*Published in:*
Biomedical Signal Processing and Control

*DOI:*
10.1016/j.bspc.2016.10.016

Published: 01/03/2017

*Document Version*
Peer reviewed version

# Matching Pursuit-Based Compressive Sensing in a Wearable Biomedical Accelerometer Fall Diagnosis Device

Ryan M. Gibson[a,b,*], Abbes Amira[a,c], Naeem Ramzan[a], Pablo Casaseca-de-la-Higuera[a], Zeeshan Pervez[a]

[a] *University of the West of Scotland, Paisley, Scotland.*
[b] *Present Address: Glasgow Caledonian University, Glasgow, Scotland.*
[c] *Qatar University, Doha, Qatar.*

**Abstract**

There is a significant high fall risk population, where individuals are susceptible to frequent falls and obtaining significant injury, where quick medical response and fall information are critical to providing efficient aid. This article presents an evaluation of compressive sensing techniques in an accelerometer-based intelligent fall detection system modelled on a wearable Shimmer biomedical embedded computing device with Matlab. The presented fall detection system utilises a database of fall and activities of daily living signals evaluated with discrete wavelet transforms and principal component analysis to obtain binary tree classifiers for fall evaluation. 14 test subjects undertook various fall and activities of daily living experiments with a Shimmer device to generate data for principal component analysis-based fall classifiers and evaluate the proposed fall analysis system. The presented system obtains highly accurate fall detection results, demonstrating significant advantages in comparison with the thresholding method presented. Additionally, the presented approach offers advantageous fall diagnostic information. Furthermore, transmitted data accounts for over 80% battery current usage of the Shimmer device, hence it is critical the acceleration data is reduced to increase transmission efficiency and in-turn improve

*Corresponding Author
*Email address:* ryan.gibson@gcu.ac.uk (Ryan M. Gibson)

battery usage performance. Various Matching pursuit-based compressive sensing techniques have been utilised to significantly reduce acceleration information required for transmission.

*Keywords:* Fall detection, compressive sensing, multiresolution analysis, principal component analysis, acceleration signal evaluation, wearable device

## 1. Introduction

Studies demonstrate elderly individuals pose a significant risk to falling [1, 2], in addition to individuals with medical conditions, for example, gait [3] and neurodegenerative disorders, including epilepsy [4] and Parkinson's disease [5]. These high-risk individuals are significantly susceptible to obtaining severe injuries from fall occurrence, including severe tissue damage and broken bones [6]. Sustained injuries are of high-importance as fall related injuries are a leading cause of death among elderly individuals [7, 8], while individuals who suffer from epilepsy are highly susceptible to significant brain injury during epileptic seizures [9]. Furthermore, these individuals may live alone as is frequent with elderly individuals and could become unable to call for help or seek medical aid due to sustained injuries during fall occurrence. Additionally, posttraumatic stress has been linked to individuals after fall occurrence [10], where adults over the age of 65 suffer additional psychological depression and stress, significantly contributing to a lower quality of life [11].

A fall detection system can be utilized to detect fall occurrence, its intensity, direction of impact and quickly raise the alarm for medical treatment if required. Additionally, an automatic fall detection system could lead to reduced fall related psychological stress and less severe head injuries occurring during epileptic seizures as the individual will receive quick medical aid appropriate to the type of fall detected.

Accelerometers can be used to automatically detect falls through different signal evaluation techniques [12]. Wearable biomedical tri-axial accelerometer-based fall detection devices transmit monitored data to a base station receiver

for signal processing and fall detection. Accelerometer-based fall detection systems frequently detect fall occurrences through thresholding a parameter, such as absolute acceleration magnitude [13] or wavelet acceleration sum-vector [14] against an arbitrary value. This arbitrary threshold value within literature is frequently determined from analysis of preknown user-sepcific fall signals [15], where accuracy and false positives are significant issues [16]. Intelligent fall detection systems utilising classifiers have more recently been utilised for accelerometer-based fall detection [17]. Accelerometer features including the raw acceleration signal [18] and acceleration signal characteristics [19] have been evaluated for fall acceleration signal classification. Various intelligent machine learning methods have been investigated for accelerometer-based fall detection applications, including k-nearest neighbours [20], artificial neural networks [21] classifiers and more recently combined classifier systems [22].

Small, low-cost and wearable biomedical embedded computer systems and devices such as the wireless Shimmer device [23] with tri-axial accelerometers, electrocardiography and gyroscope sensors are highly suited for various wearable biomedical and health implementations, such as fall evaluation. Shimmer devices have previously been utilised to detect falls with a reconfigurable high-throughput fall detection system [24, 14]. However, wearable accelerometer-based fall detection systems have been reported to obtain low-accuracy results [12] in addition to platform power constraints. The Shimmer biomedical device processor, sensors and Bluetooth radio require 5 mA, 4.89 mA and 45 mA current consumption respectively during active use, where the processor has a maximum possible current consumption of 6 mA. Furthermore, the consumption of total current utilised during active device operation demonstrates the processor and sensors to account for 9.11% and 8.91% of current consumption, while Bluetooth radio utilises a vastly significant 81.98% of total current consumption. Therefore it is critical, efficient data transmission occurs to improve power efficiency. Compressive sensing techniques [25] can be utilised to significantly reduce acceleration data transmission from the Shimmer device. Furthermore, compressive sensing has been utilised within accelerometer-based

fall detection methods to reduce the required acceleration data [26] in addition to significantly improving the Shimmer devices battery life through decreasing data transmission power requirements [27]. However, compressive sensing has only been utilised and evaluated in threshold-based systems, where currently literature indicates there has been no evaluation of compressive sensing applied to accelerometer-based features for intelligent fall detection systems.

Compressive sensing is utilised to obtain a solution from an underdetermined linear system through the advantage of signal sparsity [25]. Prior knowledge that the data are compressible allows for data to be acquired with fewer measurement samples than required with standard Nyquist-Shannon measurement sampling [28] through measuring a sparse signal of minimum samples that contain significant signal information. The resulting reduced number of samples required for signal representation is proportional to the desired compression rate and therefore obtains significant reduction in the number of measurement samples required. The corresponding recovery process of sparse signals generally involves computationally intensive and greedy algorithms which iteratively build an approximated signal solution through updating signal parameters every cycle iteration during the signal recovery process. Greedy algorithms typically do not provide the optimum solution to signal recovery, however they can outperform typical high-resolution accuracy of least-square solutions [29].

Matching pursuit (MP) [30] is a greedy algorithm that selects a suitable vector element to remove from a measurement matrix of the received sparse signal, then in each following iteration another suitable element is selected until an approximated signal is recovered. Orthogonal MP (OMP) [29] is an improvement of MP with reduced iterations at the cost of increased complexity, where OMP selects an optimal index vector element and updates the subspace columns every iteration cycle. OMP updates the subspace column vectors through orthogonally projecting the measurement vector. Other greedy sparse signal recovery algorithms have been proposed, which are often derived from OMP. MP and OMP select a single index vector column during each iteration cycle, resulting in an $M$-sparse vector requiring a minimum of $M$ iterations for signal approx-

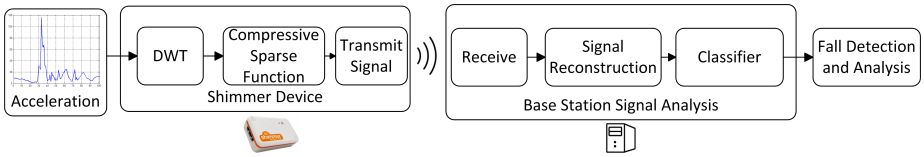



Figure 1: Simplified diagram of the proposed compressive sensing accelerometer-based intelligent fall detection system.

imation. Stagewise OMP (StOMP) [31] improves upon this limitation through being able to select multiple columns during each iteration step. Similarly, regularized OMP (ROMP) [32] is another multiple column selection sparse signal recovery algorithm, however ROMP selects vectors with a similar magnitude value.

The proposed compressive sensing accelerometer-based intelligent fall detection system implemented on a Shimmer biomedical embedded device is presented in Fig. 1, where the system's intelligent component is derived from analysis of a fall signal database to generate a fall detection and analysis classifier. The proposed system captures tri-axial acceleration signals from the wearable Shimmer device sensors. The acceleration signals are then transformed to the wavelet domain and made sparse within the Shimmer embedded computing device, before being transmitted for fall signal analysis. The received sparse signal is recovered through compressive sensing techniques to obtain the wavelet acceleration signal. The recovered wavelet signal is then classified, based on previous fall and activities of daily living (ADL) signals to determine fall occurrence, strength and direction. The presented article is a Matlab-based investigation, where the fall and ADL data obtained with the Shimmer device are recorded in Matlab. The wavelet transform and sparsification stages proposed for Shimmer device implementation, in addition to the base stations sparse signal recovery and fall analysis stages are modelled and investigated within Matlab.

The contributions of the work presented in this paper describes an intelligent, efficient and robust fall detection and diagnostic system utilising compressive

sensing with the wearable Shimmer accelerometer device. The presented fall detection system implements previously evaluated ADL and fall acceleration signals to produce robust classifiers for determining fall occurrence, strength-type (hard and soft classes) and direction. Additionally, the intelligent fall detection system is evaluated against threshold-based fall detection to determine the proposed systems advantages. Furthermore, the presented fall detection system utilises compressive sensing to significantly improve data transmission efficiency within the Shimmer accelerometer device. Literature indicates compressive sensing with accelerometer-based fall detection has currently only been investigated within threshold-based techniques [26, 27]. The effect of various compressive sensing techniques have been investigated to determine the effect on the intelligent classifier-based fall detection system results and performance.

The importance of fall monitoring and the user-wearable Shimmer device for accelerometer-based fall detection have been discussed. Additionally, the proposed system for efficient fall detection and analysis with the importance of compressive sensing for transmission efficiency have been introduced and briefly discussed in Section 1, while the rest of the paper is organised as follows: Section 2 describes the experimental set-up and acceleration data collection, Section 3 details standard magnitude threshold fall detection and principal component analysis (PCA) application for feature selection and analysis together with decision tree-based fall detection and analysis. Section 4 describes the sparse signal generation and compressive sensing of the presented fall detection system. Experimental results are presented in Section 5, while Section 6 discusses the robust and efficient fall analysis system realisation and obtained results.

## 2. Methodology

Fall and ADL acceleration data were obtained from an experimental test group of $n$ healthy subjects with differing weight and height. The test group undertook fall-based activities onto a protective region with a padded mat. Additionally, the test group undertook real-world-based ADL tasks. All subjects

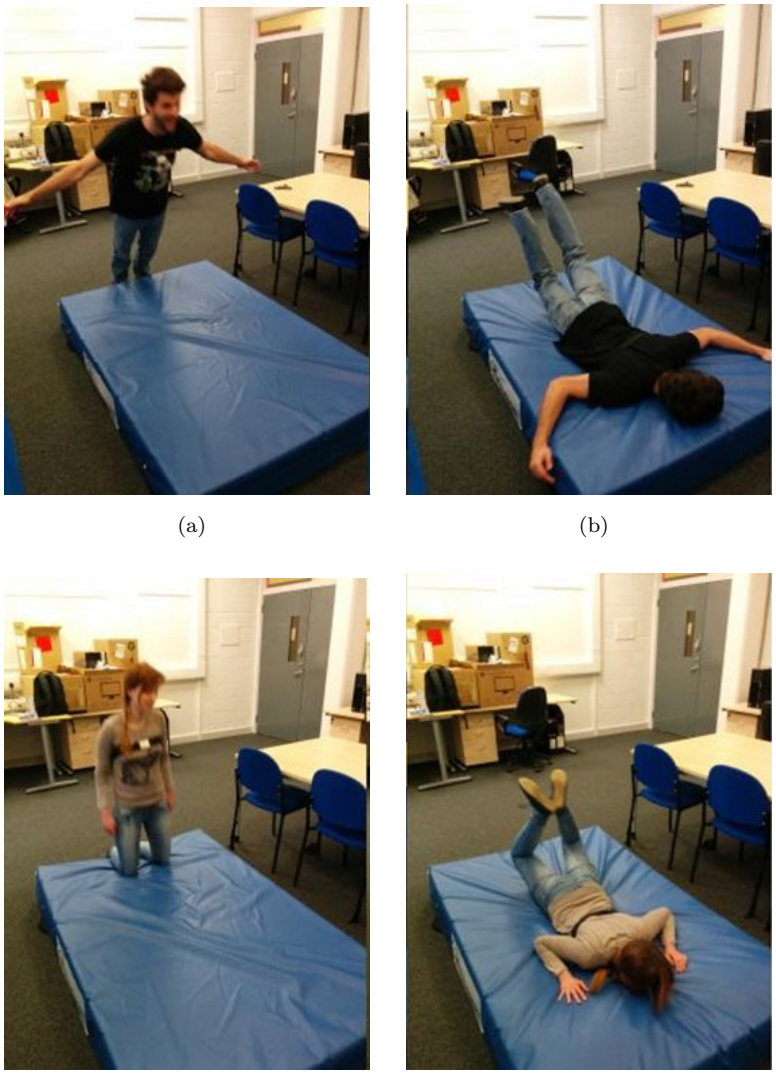
performed the required tasks, where no injuries occured. The experimental test subject group consisted of 14 individuals: 2 female and 12 male. The subject test group data for age, weight and height with mean-average and standard deviation are demonstrated in Table 1.

The Shimmer biomedical device was fastened to the subjects chest in the same position across all test subjects and throughout the various experimental fall and ADL activities performed. The fall activities performed with test subjects consisted of hard and soft impacts. Fall impact type examples are presented in Fig. 2. Hard impact falls consist of subjects falling from standing position, demonstrated in Fig. 2.*a*, where subjects would not attempt to break their fall and impact directly with the ground as demonstrated in Fig. 2.*b*. Similarly, soft impact falls consist of subjects falling from standing position, where subjects would attempt to break their fall before impact, such as falling onto their knees, as demonstrated in Fig. 2.*c*, before impacting the ground as shown in Fig. 2.*d*. Furthermore, subjects carried out impact type falls in multiple directions; left, forward, right and backwards.

Additionally, ADL tasks were undertaken with test subjects, including walking, sitting on a chair from standing position, standing up from chair, jumping and running. The ADL and fall databases are presented in Tables 2 and 3, detailing the activity type and the number of events obtained. Each subject performed every ADL type activity at least once to produce a minimum of 20 events within the data base. Walking ADL activities are a significantly frequent activity, hence double the amount of samples were recorded to obtain 40 monitored events. Furthermore, every subject produced at least one soft and hard

Table 1: Subject test group data ($n$=14).

| | Mean | Standard Deviation |
|---|---|---|
| Age | 27 | 8 |
| Weight (kg) | 80 | 20 |
| Height (cm) | 176 | 16 |



(a) (b)

(c) (d)

Figure 2: Fall type occurrences; (a) hard impact type fall with (b) straight fall to ground and (c) soft impact fall type with preliminary fall to subjects knee before (d) falling to ground.

Table 2: ADL database.

| ADL Type | Number of ADLs |
|---|---|
| Walking | 40 |
| Sitting on Chair | 20 |
| Standing Up | 20 |
| Jumping | 20 |
| Running | 20 |
| ADL Total | 120 |

Table 3: Fall database.

| Fall Type | Number of Falls |
|---|---|
| Hard Impact | 80 |
| Soft Impact | 80 |
| Left Direction | 40 |
| Forward Direction | 40 |
| Right Direction | 40 |
| Backward Direction | 40 |
| Fall Total | 160 |

impact falls in every direction to obtain 40 recorded fall events for left, forward, right and backward directions. Every fall direction-based occurrence produced 50% soft and hard impact fall types, producing 80 soft and 80 hard fall impact events. In summary, 14 test subjects obtained a database of 120 ADL and 160 fall recorded occurrences.

The obtained acceleration data from the Shimmer device was wirelessly recorded and analysed in Matlab. The Shimmer acceleration data was then processed and evaluated to model compressive sensing on the Shimmer platform and evaluate fall analysis on the received acceleration data features with the proposed fall detection system demonstrated in Fig. 3.

The proposed compressive sensing accelerometer-based intelligent fall detection system records acceleration signals in 3 dimensions (3D) on the Shimmer

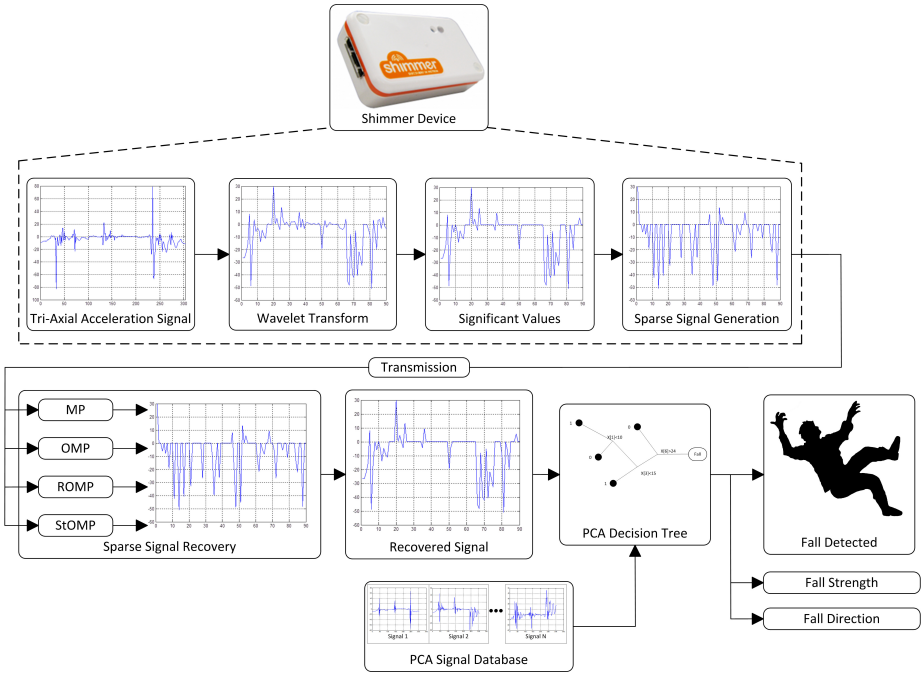



Figure 3: Presented fall detection system methodology.

device. These signals are joined together to form a tri-axial acceleration signal which is then transformed to the wavelet domain. The wavelet signal is thresholded to maintain the largest significant coefficient values while zeroing the other coefficients to obtain a sparse signal. The signals sparsity is improved through applying a preknown random permutation index function to ensure samples are randomly located throughout the vector. The sparse wavelet signal is then projected onto a random sensing matrix before being transmitted from the Shimmer device to the base station receiver for signal analysis. The received signal is reconstructed through various compressive sensing algorithms, MP, OMP, ROMP and StOMP. The reconstructed signal is then restructured in accordance with the preknown sparsity permutation function before being classified with the PCA-based decision tree to determine fall occurrence, fall direction and fall strength. The PCA-based decision tree classifier is generated

from analysis of fall and ADL signals evaluated in the wavelet domain.

## 3. Fall Signal Analysis

### *3.1. Threshold-Based Fall Detection*

Current standard approaches to accelerometer-based fall detection systems utilise a wavelet-based threshold [33, 24, 26]. The discrete wavelet transform (DWT) for a Daubechies 2 level-3 mother-wavelet is applied to the time domain tri-axial $x$, $y$ and $z$-axes acceleration components to determine the wavelet equivalent acceleration components $\alpha_x$, $\alpha_y$ and $\alpha_z$. The wavelet-domains module of acceleration vectors determine the wavelet acceleration sum vector $SV$ as follows:

$$SV = \sqrt{(\alpha_x)^2 + (\alpha_y)^2 + (\alpha_z)^2} > \tau \tag{1}$$

The threshold parameter $\tau$ is applied to the wavelet domains sum vector, where values exceeding the threshold parameter determine fall occurrence. The threshold parameter is typically selected in the literature [12] by an arbitrary value based on evaluation of fall data, however an optimum threshold $O\tau$ method has been presented [14], as follows:

$$O\tau = \frac{1}{n}\sum_{i=1}^{n} \rho_i \tag{2}$$

The optimum threshold is obtained through evaluating the average minimum fall acceleration values for each subject $\rho$ within a group of $n$ subjects, where each subjects minimum fall acceleration value is represented as $\rho_1, \rho_2 ... \rho_n$.

### *3.2. PCA-Based Smart Fall Detection Classifiers*

The presented smart fall detection system evaluates the fall and ADL database of wavelet feature coefficients feeding a binary decision tree after PCA extraction. Initial signal features are calculated using the Daubechies 2 level-3 DWT to obtain $\alpha_x$, $\alpha_y$ and $\alpha_z$ wavelet versions of the acceleration signals, $\hat{\alpha}_x$, $\hat{\alpha}_y$ and $\hat{\alpha}_z$. ADL and fall samples were obtained from 2-second windows in the time domain, yielding an overall number of 101 three dimensional vectors ($f_s = 50Hz$),

which were reorganized to vector length 303 prior to PCA calculation so that wavelet acceleration vectors $\hat{\alpha} = [\hat{\alpha}_z, \hat{\alpha}_y, \hat{\alpha}_z]$ of length 90 entered the PCA stage.

PCA is an unsupervised linear transformation method frequently utilized for data classification and reduced dimensionality. PCA utilizes statistical orthogonal functions to reduce signal dimensionality of the large observed data variables to obtain smaller significant data feature variables [34]. Let the training data set of wavelet accleration signals $\hat{\alpha}_\omega = [\hat{\alpha}_1, \hat{\alpha}_2, ...\hat{\alpha}_N]$, then the training set average is defined as $\overline{\rho} = \frac{1}{N}\sum_{i=1}^{N} \hat{\alpha}_{\omega_i}$. The wavelet acceleration signals differ from the mean-based average through $\Delta_{\hat{\alpha}_\omega} = \hat{\alpha}_{\omega_i} - \overline{\rho}$. The set of wavelet acceleration signal mean-based differences $\Delta_{\hat{\alpha}_\omega}$ are evaluated to obtain the PCA covariance matrix $CM = \frac{1}{N}\sum_{i=1}^{N} \Delta_{\hat{\alpha}_{\omega_i}} \Delta^{\Gamma}_{\hat{\alpha}_{\omega_i}}$, where the orthogonal counterpart of the wavelet acceleration signal mean-based difference is denoted as $\Delta^{\Gamma}_{\hat{\alpha}_\omega}$. The covariance matrix is further evaluated with eigendecomposition to determine the associated eignenvalues $\lambda_k$ and eigenvectors $u_k$. The eigenvalue data vector is arranged in descending order, where $M$ principle components are selected $\lambda_1, \lambda_2, ...\lambda_M$. The PCA implementation presented utilizes $M = 45$ principle components to determine the feature subspace dimensionality through obtaining a projection matrix $W$. Unsupervised linear transformation with PCA is determined with $\hat{\alpha}^{PCA}_{\omega} = W^{\Gamma}\hat{\alpha}_\omega$ The reduced signal dimensionality output $\hat{\alpha}^{PCA}_{\omega}$ of the PCA module was classified using binary decision trees [35] trimmed from 45 predictors to 26 for detecting falls and determining their strength and direction. The decision tree functions were thus trained using PCA coefficients $\hat{\alpha}^{PCA}_{\omega}$ for fall detection, strength and direction.

## 4. Compressive Sensing

Compressive sensing allows an input signal in high-dimensional space to be compressed to a signal of significantly smaller dimensions, such as the wavelet-domain [25, 36]. The acceleration signal in the time-domain $\Gamma$ of $N$ samples is decomposed with an orthonormal Daubechies 2 wavelet basis $\Psi = [\psi_1|\psi_2|\ldots|\psi_N]$ to obtain a compressive sensing suitable sparse wavelet-domain coefficient signal

$X$ as shown:

$$X = \Psi\Gamma \tag{3}$$

The presented sparse wavelet signal undergoes a sparsity function $\mu$ to improve sparsity properties with thresholding and preknown indexing. The compressive sensing function can be described as follows:

$$Y = \Phi X = \Phi\mu\Psi\Gamma \tag{4}$$

Where the input signal vector $X$ of length $N$ is $K$-sparse, $Y$ is the observation vector of length $M$ and the Gaussian sensing matrix $\Phi$ is of size $M \times N$. Compressive sensing is significantly suited for $K$-sparse signals, where signal vector $X$ of length $N$ is represented by significant $K$ nonzero coefficients [37]. The input signal $X$ is sparse when only a low number of coefficients are nonzero. Sensing matrices are frequently implemented as random matrix functions with independent distribution such as Gaussian or Bernoulli distributions [38], which have previously been demonstrated to provide efficient signal probability recovery [39]. Additionally, compressive sensing captures $M$ signal measurements from $N$ signal samples through random linear projections [39, 25], where $M << N$, offering advantages for significantly improved data efficiency suitable for power-limited transmission platforms.

The Shimmer platforms tri-axial acceleration signals are merged to a single vector of length $N = 303$ demonstrated in Fig. 4.*a*, a Daubechies 2 level-3 DWT is applied, obtaining a wavelet feature and detail signal $N = 90$ as shown in Fig. 4.*b*, suitable for fall analysis with the presented intelligent PCA-based fall detection classifiers. The wavelet signals sparse properties are further improved through applying the sparsity function $\mu$. Where the wavelet signal is thresholded to obtain a wavelet signal vector $N = 90$, retaining $K = 45$ largest nonzero value coefficients, before applying a preknown random index permutation function as shown in Fig. 4.*c*. Once the wavelet acceleration signal has been sparsed, the compression described in (4) is carried out, where a Gaussian distribution is implemented as a sensing matrix. The compressed sparse wavelet acceleration signal $Y$ is then transmitted. The greedy sparse

signal recovery algorithms MP, OMP, ROMP and StOMP previously described are applied to recover the estimated thresholded wavelet acceleration signal $\hat{X}$. The known index permutation function is applied and the wavelet coefficients $N = 90$ are extracted as demonstrated in Fig. 4.*d* and applied to the PCA classifiers described previously in Section 3.2. The presented technique exploits the received compressed wavelet acceleration data to perform PCA-based fall detection classification without performing the inverse DWT.

MP is a technique frequently utilised for sparse signal reconstruction through a greedy iterative process to compute an approximation of the original signal

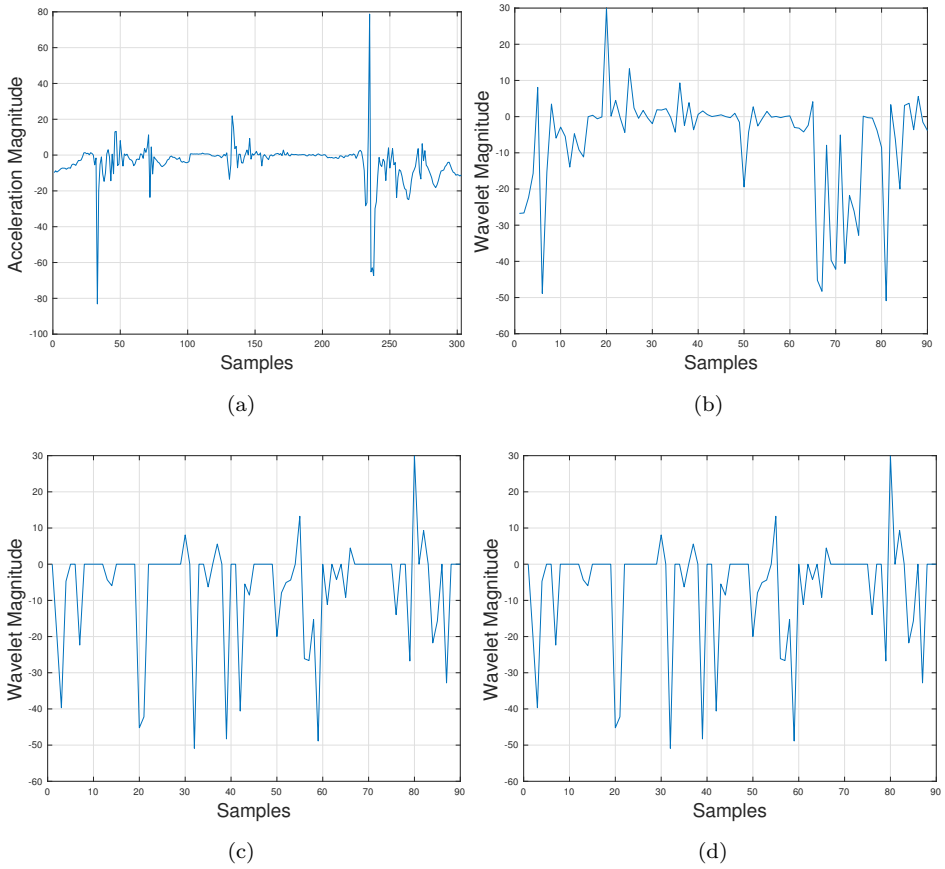



Figure 4: Fall example of (a) acceleration, (b) wavelet-domain signal, (c) thresholded with random permutation and (d) OMP estimate.

as demonstrated in Algorithm 1. The MP algorithm iteratively determines the single most suitable vector element to remove from the measurement matrix $\Phi$ and repeats the subtraction process from the residual component $r$ during each iteration until the estimated original signal $\hat{X}$ is determined. It is important to note that MP will repeatedly select the same vector element to remove from the sensing matrix in order to improve the estimated signal. MP is a relatively low-complexity solution suited to high-sparsity signals [30]. Similarly, OMP determines an estimated signal through an iterative process shown in Algorithm 2. In OMP, the approximated signal $\hat{X}$ is updated every iteration with projecting $Y$ orthogonally onto columns of the sensing matrix $\Phi$ for the current index set $\Lambda$. OMP never selects a previously evaluated vector element from the sensing matrix and the residual component during any iteration is orthogonal to currently selected vector elements. OMP is a more complex technique in comparison to MP, as least squares is utilised to estimate the signal during every iteration, however it reduces iterations required for signal estimation. MP and OMP methods need to perform a minimum iteration count determined by the number of elements to be selected, which does not scale well for large dimension signals. ROMP and StOMP algorithms shown in Algorithms 3 and 4 are multiple-element selection-based techniques derived from OMP and utilise an updated residual component every iteration until signal estimation is met. ROMP groups inner elements $\lambda$ into sets $\Lambda$, where elements in each set contain a similar magnitude [40]. StOMP groups inner elements $\lambda$ above a threshold parameter $\tau$ into sets $\Lambda$. The process is then repeated on the residual vector until the estimated signal is obtained. StOMP can generally obtain good approximations through few iterations through evaluating multiple vectors [41]. For direct comparison all reconstruction algorithms were configured to operate for $K$ or less iterations.

**Algorithm 1** MP algorithm for signal recovery.

**Input:** $\Phi$: Sensing matrix; $Y$: Observation vector; $K$: Sparsity of signal $X$.

**Output:** $\hat{X}$: Original signal estimation.

***Procedure***:

1: Initialise residual $r_0 = Y$ and set iteration counter $i = 1$.

2: Find index $\lambda_i = \arg\max |\langle r_{i-1}, \emptyset_j \rangle|$ for $j = 1 \dots N$.

3: Calculate new approximation $\hat{X}_i = \hat{X}_{i-1} + \langle \Phi_{\lambda_i}, r_{i-1} \rangle \Phi_{\lambda_i}$ and residual $r_i = Y - \hat{X}_{i-1} = r_i - \langle \Phi_{\lambda_i}, r_{i-1} \rangle \Phi_{\lambda_i}$.

4: Increment iteration counter $i = i + 1$, return to step 2 if $i < K$.

5: Output estimate $\hat{X}$.

**Algorithm 2** OMP algorithm for signal recovery.

**Input:** $\Phi$: Sensing matrix; $Y$: Observation vector; $K$: Sparsity of signal $X$.

**Output:** $\hat{X}$: Original signal estimation.

***Procedure***:

1: Initialise residual $r_0 = Y$ and index set $\Lambda_0 = \{\emptyset\}$. Iteration counter $i = 1$.

2: Find index $\lambda_i = \arg\max |\langle r_{i-1}, \emptyset_j \rangle|$ for $j = 1 \dots N$.

3: Increase index set $\Lambda_i = \Lambda_{i-1} \cup \{\lambda_i\}$ and matrix column vectors $\Phi_i = [\Phi_{i-1}\ \Phi_{\lambda_i}]$.

4: Solve least squares problem for estimate $\hat{X}$ of signal $X$; $\hat{X}_i = \arg\min \|Y - \Phi_i \hat{X}_i\|$.

5: Calculate new residual $r_i = Y - \Phi_i \hat{X}_i$.

6: Increment iteration counter $i = i + 1$, return to step 2 if $i < K$.

7: Output estimate $\hat{X}$.

## 5. Results

Recall $\phi$, precision $\varphi$ and specificity $\delta$ metrics as shown in (5) through (7) can be utilised to evaluate fall detection, direction and strength results obtained with the proposed system.

$$\phi = \frac{T_p}{T_p + F_n} \tag{5}$$

**Algorithm 3** ROMP algorithm for signal recovery.

**Input:** $\Phi$: Sensing matrix; $Y$: Observation vector; $K$: Sparsity of signal $X$.

**Output:** $\hat{X}$: Original signal estimation.

***Procedure***:

1: Initialise residual $r_0 = Y$ and index set $\Lambda_0 = \{\emptyset\}$. Iteration counter $i = 1$.

2: Identify set $\lambda_i$ of $K$ largest coordinates in observation vector $u_i = \Phi_i r_i$.

3: Regularise subsets $\lambda_0 \subset \lambda_i$ with comparable coordinates $|u_i| \leq 2|u_{\lambda_i}|$ for $i, \lambda \in \lambda_0$.

4: Update index set $\Lambda_i = \Lambda_{i-1} \cup \lambda_0$.

5: Calculate new approximation $\hat{X}_i = \arg\min \|Y - \Phi_i \hat{X}_i\|$ and residual $r_i = Y - \Phi_i \hat{X}_i$.

6: Increment iteration counter $i = i + 1$, return to step 2 if $i < K$.

7: Output estimate $\hat{X}$.

**Algorithm 4** StOMP algorithm for signal recovery.

**Input:** $\Phi$: Sensing matrix; $Y$: Observation vector; $K$: Sparsity of signal $X$.

**Output:** $\hat{X}$: Original signal estimation.

***Procedure***:

1: Initialise residual $r_0 = Y$ and index set $\Lambda_0 = \{\emptyset\}$. Iteration counter $i = 1$.

2: Obtain residual correlation vector $c_s$ with matched filter, $c_s = \Phi_i^T r_{i-1}$.

3: Threshold correlation residual with $\tau$ to create index $\lambda_i$.

4: Update index set $\Lambda_i = \Lambda_{i-1} \cup \lambda_i$.

5: Calculate new approximation $\hat{X}_i = \arg\min \|Y - \Phi_i \hat{X}_i\|$ and residual $r_i = Y - \Phi_i \hat{X}_i$.

6: Increment iteration counter $i = i + 1$, return to step 2 if $i < K$.

7: Output estimate $\hat{X}$.

310

$$\varphi = \frac{T_p}{T_p + F_p} \tag{6}$$

$$\delta = \frac{T_n}{T_n + F_p} \tag{7}$$

Recall, precision and specificity are obtained from true and false positive and negative rates: $T_p$, $F_p$, $T_n$, $F_n$. The confusion matrix defining these rates is presented in Table 4 for events and non-events detected by the system:

The events and non-events are defined depending on the measured variable:

- Fall detection: Falls are defined as events and ADLs as non-events.
- Fall strength: For hard falls, a hard fall is defined as an event. ADLs and soft falls are counted among non-events. For soft falls, the procedure is inverted, being hard falls part of the non-events, and soft ones the actual events. The overall metrics for fall strength are presented as an average of both hard and soft falls.
- Fall direction: For each of the directions considered (forward, backward, left, and right directional falls), falls in that direction are the actual events, while falls in the remaining directions and ADLs are non-events. The overall metrics are computed averaging all the directions.

The system's recall value accounts for the actual sensitivity to events, while precision is a measurement of the predictive positive value for the events. Finally, specificity measures the robustness of the system against false positives (i.e., identifying non-events as events). An overall accuracy measure for the system can be obtained from the F-value $f$ metric, calculated as the harmonic mean of recall and precision:

$$f = 2\frac{\phi\varphi}{\phi + \varphi}. \tag{8}$$

Table 4: Confusion matrix defining $T_p$, $F_p$, $T_n$, $F_n$ for events and non-events detected by the system

| | Predicted Event | Predicted Non-event |
|---|---|---|
| Actual Event | $T_p$ | $F_p$ |
| Actual Non-event | $F_n$ | $T_n$ |

In this article, PCA-based decision tree classifiers are generated from a training group and evaluated against a control group in accordance with k-fold cross-validation techniques [42] for 10-fold and 5-fold methods. The 10-fold cross validation technique seperates the signal data base into 10 equal sized groups, where the PCA classifier is trained utilizing 90% of acceleration signal data, while the remaining 10% is utilized for testing. Similarly, 5-fold cross-validation seperates the data base into 5 equal sized groups, where 80% and 20% of data is utilized for training and testing of the PCA classifier. Furthermore, the process is repeated throughout the data base groups, where each instance of testing is validated against the remaining groups. This obtains a robust testing validation as the data base is evaluated with different combinations for testing and training samples, where no sample appears simultaneously within training and testing groups. The obtained results for 10-fold and 5-fold cross-validation methods were recorded, where an average for each method is presented. Threshold-based fall detection with the optimum threshold parameter [14] and PCA classifier-based fall detection, strength and direction results for recall, precision, specificity and F-value without compressive sensing for 10-fold and 5-fold cross-validation methods are presented in Tables 5 and 6.

The optimum threshold method demonstrated accuracy of 87% recall, 93% precision, 81% specificity and 90% F-value, while the presented PCA Classifier method obtained significant accuracy improvements of 9%, 6%, 17% and 7% for recall, precision, specificity and F-Value respectively, obtaining 96% recall, 99% precision, 98% specificity and 97% F-value for 10-fold cross-validation.

Table 5: Fall detection results with 10-fold cross validation

| Fall Analysis | Recall (%) | Precision (%) | Specificity (%) | F-Value (%) |
|---|---|---|---|---|
| Threshold-Based Fall Detection | 87 | 93 | 81 | 90 |
| Classifier-Based Fall Detection | 96 | 99 | 98 | 97 |
| Classifier-Based Fall Strength | 91 | 99 | 96 | 95 |
| Classifier-Based Fall Direction | 87 | 98 | 84 | 93 |

Table 6: Fall detection results with 5-fold cross validation

| Fall Analysis | Recall (%) | Precision (%) | Specificity (%) | F-Value (%) |
|---|---|---|---|---|
| Threshold-Based Fall Detection | 76 | 81 | 71 | 78 |
| Classifier-Based Fall Detection | 86 | 90 | 89 | 88 |
| Classifier-Based Fall Strength | 82 | 91 | 90 | 86 |
| Classifier-Based Fall Direction | 83 | 90 | 79 | 86 |

Additionally, the presented smart PCA classifier demonstrated 91% recall, 99% precision and 95% F-value for fall strength accuracy and 87%, 98% and 93% F-value for fall direction with 10-fold cross-validation. The optimum threshold method demonstrated consistent performance decreases of approximately 10% as testing and training data increased and decreased simultaenously in accordance with 5-fold cross-validation. In contrast PCA-based classifier techniques demonstrated a range of performance decreases from 4% to 10% in comparison with 10-fold cross validation obtained results. Hence, the PCA-based classifier method is demonstrated to be more robust to reduced training data with k-fold cross validation techniques than the threshold fall detection method. The rest of this article will present, evaluate and describe the 10-fold cross-validation implementation.

Multiple compressive sensing techniques were applied to the proposed system as demonstrated in Fig. 3. The sparse received signal with $K = 45$ nonzero coefficient elements over vector length $N = 90$ is reconstructed with MP, OMP, ROMP and StOMP compressive sensing techniques. The reconstructed signals are evaluated against signal sparsity as defined in (9), where the number of significant nonzero coefficients $K$ are reduced in the transmitted signal vector length $N$ to increase sparsity.

$$Sparsity(\%) = \left(1 - \frac{K}{N}\right) * 100 \tag{9}$$

The percentage of exact coefficient recovery for MP, OMP, ROMP and StOMP compressive sensing techniques evaluated against sparsity increase are presented in Fig. 5.

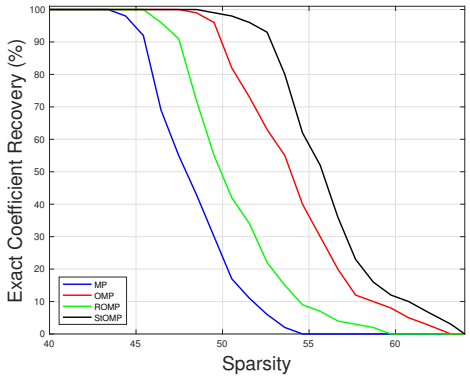



Figure 5: Percentage of exact coefficient recovery as sparsity increases.

The reconstructed signal initially recovers 100% exact coefficients until sparsity approaches 43%, 46%, 48% and 50% sparsity for MP, ROMP, OMP and StOMP respectively, where the percentage of exact coefficient recovery begins to decrease. The percentage of exact coefficient recovery continues to decrease as sparsity increases until 0% exact coefficients recovery occurs at 54%, 59%, 64% and 65% sparsity for MP, ROMP, OMP and StOMP respectively.

Evaluating MP, OMP, ROMP and StOMP recovered signals with the presented PCA-based decision tree classifier in terms of recall, precision, specificity and F-value for fall detection, strength and direction as sparsity increases have been presented in Fig. 6, 7 and 8 respectively. Where sparsity increase was obtained from reducing the number of largest significant coefficient values $K$ of the transmitted sparse wavelet signal.

All sparse signal recovery algorithms were initially able to obtain the same recall, precision and F-value accuracy for fall detection, strength and direction as presented in Table 5. However the general response of recall, precision, specificity and F-value results decrease as sparsity increases from 50%, until 0% accuracy occurs at 68% sparsity. Where there would frequently be a sudden significant decrease occurring at approximately 67% sparsity towards 0% accuracy. The initial decrease of recall, precision, specificity and F-value accuracy for fall detection, strength and direction occurred at approximately the same sparsity

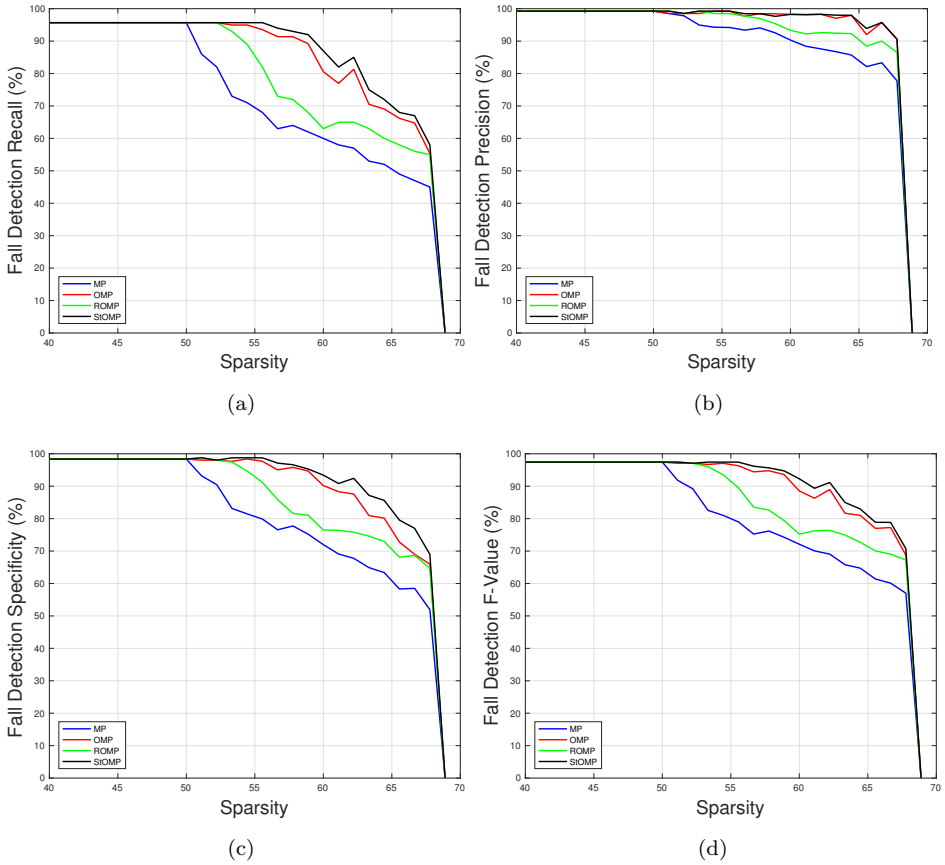



Figure 6: Fall detection (a) recall, (b) precision, (c) specificity and (d) F-value responses.

percentage for each compressive sensing technique. MP demonstrated accuracy decrease occurs at 50%, 53%, 50% and 50% sparsity for recall, precision, specificity and F-value parameters respectively. OMP demonstrated accuracy decrease occurs at 55%, 64%, 56% and 55% sparsity for recall, precision, specificity and F-value parameters respectively. ROMP demonstrated accuracy decrease occurs at 53%, 56%, 53% and 53% sparsity for recall, precision, specificity and F-value parameters respectively. StOMP demonstrated accuracy decrease occurs at 56%, 65%, 57% and 56% sparsity for recall, precision, specificity and F-value parameters respectively. However the responses after the initial decrease for each compressive sensing technique are significantly different across

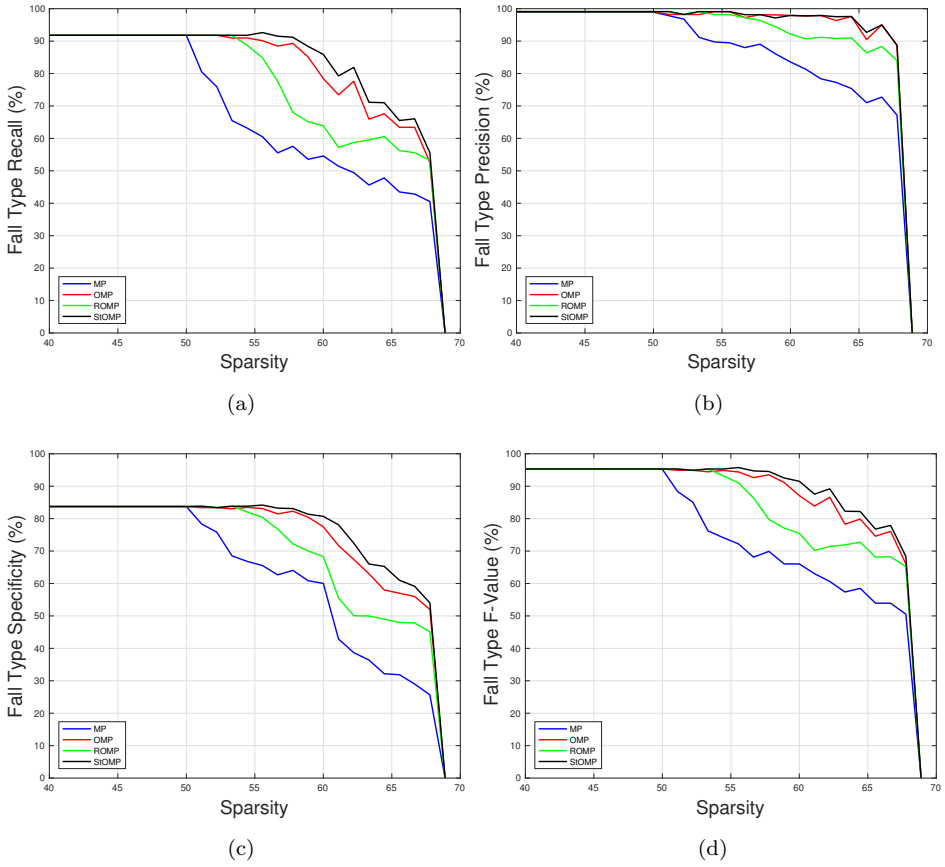



Figure 7: Fall type (a) recall, (b) precision, (c) specificity and (d) F-value responses.

fall analysis. For example, fall detection recall with MP and ROMP obtain 14% and 29% accuracy at 67% sparsity as shown in Fig. 6.*a*. While in comparison MP and ROMP obtain significantly different values of 44% and 52% accuracy for fall strength recall and 37% and 50% accuracy for fall direction with 67% sparsity as demonstrated in Fig. 7.*a* and 8.*a*.

The various compressive sensing techniques performance can be ranked from highest to lowest obtained accuracy during sparsity increase, where StOMP demonstrated the best results, followed by OMP then ROMP and the worst performance was obtained with MP. StOMP obtained the highest recall, precision, specificity and F-value metrics throughout, followed by OMP demonstrating lit-

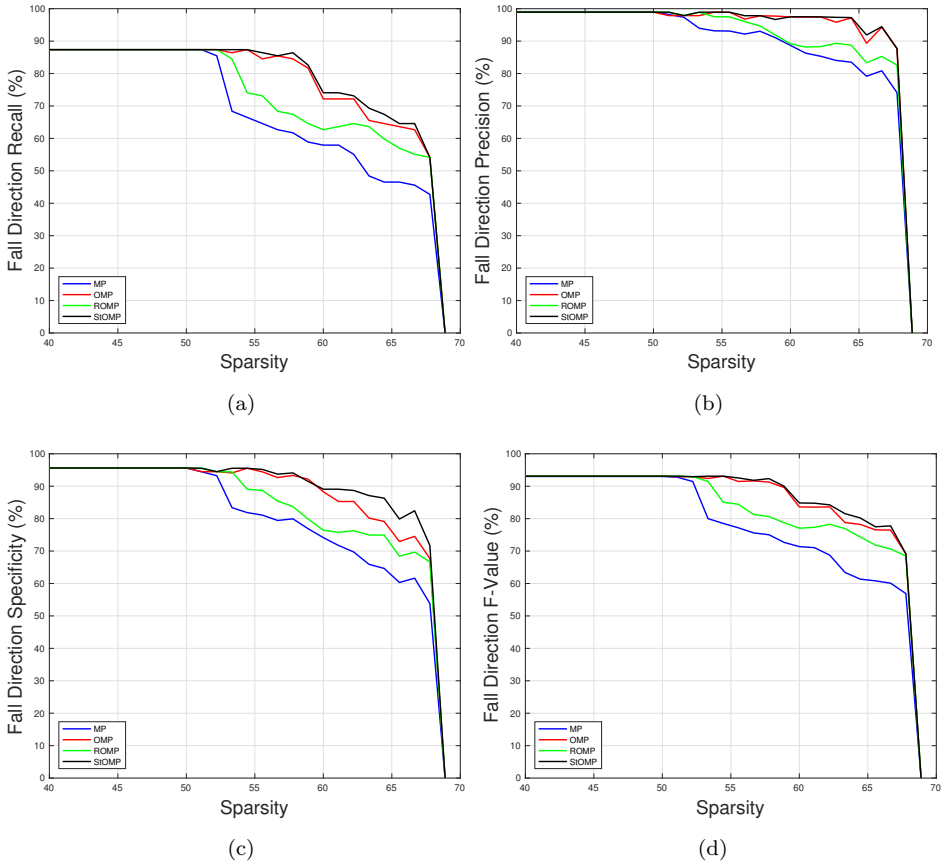



Figure 8: Fall direction (a) recall, (b) precision, (c) specificity and (d) F-value responses.

tle significant difference between response, where precision results obtained were similar, for example as demonstrated with fall detection precision as shown in Fig. 6.*b*. ROMP and MP demonstrated significantly worse results than StOMP and OMP, where MP and ROMP obtained quick and high-accuracy loss as sparsity increased. Recall, specificity and F-test accuracy demonstrated a large accuracy decrease as sparsity increased, however the classifiers precision maintained relatively high until sparsity reached approximately 65%, e.g. Fig. 8.*b*.

Additionally, the PCA-based classifier was able to obtain high-accuracy results while the equivalent exact coefficient recovery percentage could be relatively low. For example, StOMP obtained 70%, 96%, 82% and 80% accuracy

for fall detection recall, precision, specificity and F-value responses respectively at 65% sparsity as demonstrated in Fig. 6. While in comparison, the exact coefficient recovery was 0% at 65% sparsity as shown in Fig. 5, indicating the obtained signal error at 65% sparsity is within the PCA-based classifiers boundaries for correct fall analysis. Furthermore, the significant drop of recall, precision, specificity and F-value at 67% sparsity could be deduced to occur when signal error increases to outside the classifiers boundary limits for correct fall analysis. Recall, specificity and F-test accuracy generally demonstrated a large accuracy decrease as sparsity increased, however the classifiers precision metric maintained relatively high accuracy until sparsity reached approximately 65%. MP and ROMP techniques obtained the lowest performance results in comparison to OMP and StOMP methods which performed well in the presented fall evaluation application. Potential hypothesis for the different performance results include: MP has a convergence problem, where it can potentially reselect a previously selected vector removed from the signal. The convergence problem of MP has demonstrated much larger error rates in comparison to OMP [43], where OMP removes the potential vector reselection issue through updating its dictionary during each iteration. Furthermore OMP utilizes least squares to estimate signal approximation during every iteration further increasing the possibilities for improved performance. ROMP has been recorded in literature for demonstrating poor performance in application, where ROMP implementations are best suited to very large $N$ column data sets [41]. Possibly indicating the fall detection application presented does not contain data large enough over $N$ columns for high ROMP performance. StOMP obtained relatively high performance results, demonstrating a small performance increase over OMP. This may be due to the fall application utilizing Gaussian sensing matrices and StOMP being limited to theory on Gaussian matrices, where theoretical performance is guaranteed when applied to general Gaussian matrices [41]. Furthermore, the proposed system offers significant space savings defined with the compression ratio $\omega$ expressed in (10), evaluating the number of original data samples $N_{orig}$ and the number of data samples in the compressive sensing sparse signal $N_{comp}$.

$$\omega(\%) = \left( \frac{N_{orig} - N_{comp}}{N_{orig}} \right) * 100 \tag{10}$$

The original Shimmer device acceleration transmitted signal vector of length $N_{orig} = 303$, the compressed signal obtained a sparse signal vector of length $N_{comp} = 90$ with 45 nonzero elements. The compressed signal vector $N_{comp} = 90$ demonstrated a significant compression ratio of 70.3%, while the compression ratio in context of significant nonzero coefficients $N_{comp} = 45$ obtained a significant compression ratio of 85.1% in comparison to the original acceleration signal. In worst case scenario the Shimmer device will consume 6 mA, 4.89 mA and 45 mA for CPU, sensors and Bluetooth transmission respectively, where the components utilize the maximum power consumption possible. This would be representive of a Shimmer device with a highest supported sampling frequency constantly recording data from all sensors with constant Bluetooth transmission, in addition to the CPU obtaining the largest supported current usage. The Shimmer device contains a 280 mAh battery, where under this worst-case scenario standard sensing and transmission would obtain 5.4 hours of usage. In contrast, the compressive sensing techniques presented offers a maximum transmission space savings of 85.1% which would obtain 21.2 hours of usage under this worst-case scenario. The presented compressive sensing techniques demonstrate a 292% improvement in battery life in comparison to normal sensing and transmission of the Shimmer device. Investigations within literature [23] have reported real-world usage values for standard Bluetooth data streaming on a Shimmer device with accelerometer sensors sampling at 50 Hz, as used in this application, to obtain 15.9 mA average current consumption and a corresponding 17.6 hours of Shimmer usage. Applying the same percentage improvement obtained during worst case scenario provide an estimate of the compressive sensing fall detection techniques obtaining 69 hour operation on a single charge of a Shimmer device.

The presented PCA-based decision tree classifier method significantly outperformed the threshold-based fall detection method frequently utilised within

literature [33, 24, 26] for accelerometer-based fall detection systems. The PCA-based classifier obtained high-accuracy results with recall, specificity, precision and F-value for fall detection, strength and direction, where the classifier demonstrated significantly higher precision results. The sparsity of the proposed systems transmitted signal was increased through reducing the number of significant coefficients retained. StOMP overall performed better during evaluation than MP, OMP and ROMP, where StOMP and OMP obtained relatively similar results with fall detection, strength and direction classifiers. The classifiers accuracy failed as the transmitted signal vector reached 68% sparsity with all compressive sensing techniques. Furthermore, the PCA-based decision tree classifier offers significant data reduction with a compression ratio over 70% in comparison to standard acceleration signal transmission within the Shimmer device. The obtained transmission data reduction has a direct effect on significantly improving transmission efficiency with the Shimmer device.

## 6. Conclusions

This article presents a compressive sensing acceleration-based intelligent fall detection and analysis system with a Shimmer embedded computing device modelled with Matlab. The proposed system utilises fall and ADL acceleration signals within the wavelet-domain to obtain PCA-based binary decision tree fall classifiers for fall detection, strength-type and direction. Test subjects generated fall and ADL acceleration signals with the Shimmer device. The captured tri-axial acceleration signals were converted to the wavelet-domain and made sparse through selecting the most significant coefficients for wireless transmission to a base station. The received sparse signal is then recovered through greedy compressive sensing techniques: MP, OMP, ROMP and StOMP. The reconstructed signal is then evaluated with the PCA-based decision tree classifier to determine fall detection, strength and direction. The presented system obtained highly-accurate fall detection, strength and direction results for recall, precision, specificity and F-value accuracy. Acceleration-based fall detection systems

have reported low-accuracy response with threshold-based detection methods [12]. The optimum threshold method presented demonstrated 87%, 93%, 81% and 90% for fall detection recall, precision, specificity and F-value measures. The presented PCA-based decision tree technique obtained 96%, 99%, 98% and 95% for fall detection recall, precision, specificity and F-value measures respectively, demonstrating significant improvements in comparison with the optimum threshold approach. Additionally, threshold approaches are significantly limited as they are only capable of detecting fall occurrence. While in comparison the presented PCA-based approach can determine fall occurrence in addition to important fall information including whether the fall was a hard or soft impact, in addition to the direction the individual fell. Acceleration data transmission with the Shimmer device accounts for over 80% of total current consumption during standard use, indicating the importance of efficient data transmission on the limited power Shimmer platform. Various compressive sensing techniques were utilised to significantly reduce acceleration information required for transmission with the Shimmer device. StOMP obtained the best signal performance as signal sparsity increased, where 0% exact coefficient recovery occurred at 54%, 59%, 63% and 65% for MP, OMP, ROMP and StOMP techniques. MP, OMP, ROMP and StOMP were initially able to obtain the same recall, precision and F-value accuracy for fall detection, strength and direction, however as sparsity increased the accuracy decreased. Where StOMP obtained the best results for fall detection, strength and direction with OMP obtaining slightly lower accuracy results, while MP and ROMP demonstrated significantly low-accuracy results in comparison. Furthermore, the presented compressive sensing fall detection system obtains significant data reduction of over 70%, demonstrating improved transmission efficiency while maintaining the systems high accuracy results with an estimated 292% increase of Shimmer battery life.

## References


[1] H. Axer, M. Axer, H. Sauer, O. W. Witte, G. Hagemann, Falls and gait disorders in geriatric neurology, Clinical neurology and neurosurgery 112 (4) (2010) 265–274.

[2] A. Kabeshova, C. Annweiler, B. Fantino, T. Philip, V. A. Gromov, C. P. Launay, O. Beauchet, A regression tree for identifying combinations of fall risk factors associated to recurrent falling: a cross-sectional elderly population-based study, Aging clinical and experimental research 26 (3) (2014) 331–336.

[3] S. M. Rispens, K. S. van Schooten, M. Pijnappels, A. Daffertshofer, P. J. Beek, J. H. van Dieën, Identification of fall risk predictors in daily life measurements gait characteristics reliability and association with self-reported fall history, Neurorehabilitation and neural repair (2014) 1545968314532031.

[4] M. Atkinson, A. Shah, K. Hari, K. Schaefer, P. Bhattacharya, A. Shah, Safety considerations in the epilepsy monitoring unit for psychogenic nonepileptic seizures, Epilepsy & Behavior 25 (2) (2012) 176–180.

[5] C. G. Canning, S. S. Paul, A. Nieuwboer, Prevention of falls in parkinson's disease: a review of fall risk factors and the role of physical interventions, Neurodegenerative disease management 4 (3) (2014) 203–221.

[6] M. E. Tinetti, C. S. Williams, Falls, injuries due to falls, and the risk of admission to a nursing home, New England journal of medicine 337 (18) (1997) 1279–1284.

[7] P. Kannus, J. Parkkari, S. Niemi, M. Palvanen, Fall-induced deaths among elderly people, American journal of public health 95 (3) (2005) 422.

[8] C. for Disease Control, Prevention, et al., Fatalities and injuries from falls among older adults—united states, 1993–2003 and 2001–2005, MMWR: Morbidity and Mortality Weekly Report 55 (45) (2006) 1221–1224.

[9] D. A. Wilson, A. W. Selassie, Risk of severe and repetitive traumatic brain injury in persons with epilepsy: A population-based case–control study, Epilepsy & Behavior 32 (2014) 42–48.

[10] J. Williams, B. Kool, E. Robinson, S. Ameratunga, Longer term health of young and middle-aged adults following unintentional falls at home resulting in hospitalisation, Injury 43 (1) (2012) 103–108.

[11] N. Jayasinghe, M. A. Sparks, K. Kato, K. Wyka, K. Wilbur, G. Chiaramonte, P. S. Barie, M. S. Lachs, M. O'Dell, A. Evans, et al., Posttraumatic stress symptoms in older adults hospitalized for fall injury, General hospital psychiatry.

[12] M. Mubashir, L. Shao, L. Seed, A survey on fall detection: Principles and approaches, Neurocomputing 100 (2013) 144–152.

[13] L. Mao, D. Liang, Y. Ning, Y. Ma, X. Gao, G. Zhao, Pre-impact and impact detection of falls using built-in tri-accelerometer of smartphone, in: Health Information Science, Springer, 2014, pp. 167–174.

[14] R. M. Gibson, A. Amira, P. Casaseca-de-la Higuera, N. Ramzan, Z. Pervez, An efficient user-customisable multiresolution classifier fall detection and diagnostic system, in: Microelectronics, 2014. ICM 2014. 26th International Conference on, IEEE, 2014, p. in print.

[15] P. T. Phu, N. T. Hai, N. T. Tam, A threshold algorithm in a fall alert system for elderly people, in: 5th International Conference on Biomedical Engineering in Vietnam, Springer, 2015, pp. 347–350.

[16] Q. T. Huynh, U. D. Nguyen, K. T. Liem, B. Q. Tran, Detection of activities daily living and falls using combination accelerometer and gyroscope, in: 5th International Conference on Biomedical Engineering in Vietnam, Springer, 2015, pp. 184–189.

[17] C. Li, M. Lin, L. T. Yang, C. Ding, Integrating the enriched feature with machine learning algorithms for human movement and fall detection, The Journal of Supercomputing 67 (3) (2014) 854–865.

[18] N. Shibuya, B. Nukala, A. Rodriguez, J. Tsay, T. Nguyen, S. Zupancic, D. Lie, A real-time fall detection system using a wearable gait analysis sensor and a support vector machine (svm) classifier, in: Mobile Computing and Ubiquitous Networking (ICMU), 2015 Eighth International Conference on, IEEE, 2015, pp. 66–67.

[19] N. Ravi, N. Dandekar, P. Mysore, M. L. Littman, Activity recognition from accelerometer data, in: AAAI, Vol. 5, 2005, pp. 1541–1546.

[20] A. T. Özdemir, B. Barshan, Detecting falls with wearable sensors using machine learning techniques, Sensors 14 (6) (2014) 10691–10708.

[21] V. N. T. Sang, N. D. Thang, V. Van Toi, N. D. Hoang, T. Q. D. Khoa, Human activity recognition and monitoring using smartphones, in: 5th International Conference on Biomedical Engineering in Vietnam, Springer, 2015, pp. 481–485.

[22] R. M. Gibson, A. Amira, N. Ramzan, P. Casaseca-de-la Higuera, Z. Pervez, Multiple comparator classifier framework for accelerometer-based fall detection and diagnostic, Applied Soft Computing 39 (2016) 94–103.

[23] A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, V. Cionca, Shimmer–a wireless sensor platform for noninvasive biomedical research, Sensors Journal, IEEE 10 (9) (2010) 1527–1534.

[24] M. Neggazi, A. Amira, L. Hamami, A wireless reconfigurable system for falls detection, in: Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on, IEEE, 2012, pp. 77–82.

[25] D. L. Donoho, Compressed sensing, Information Theory, IEEE Transactions on 52 (4) (2006) 1289–1306.

[26] M. Neggazi, L. Hamami, A. Amira, Efficient compressive sensing on the shimmer platform for fall detection, in: Circuits and Systems (ISCAS), 2014 IEEE International Symposium on, IEEE, 2014, pp. 2401–2404.

[27] M. Neggazi, L. Hamami, A. Amira, A multi-scale analysis and compressive sensing based energy aware fall detection system, in: Design and Test Symposium (IDT), 2013 8th International, 2013, pp. 1–3.

[28] E. J. Candès, Compressive sampling, in: Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures, 2006, pp. 1433–1452.

[29] J. A. Tropp, A. C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, Information Theory, IEEE Transactions on 53 (12) (2007) 4655–4666.

[30] S. G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, Signal Processing, IEEE Transactions on 41 (12) (1993) 3397–3415.

[31] D. L. Donoho, Y. Tsaig, I. Drori, J.-L. Starck, Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit, Information Theory, IEEE Transactions on 58 (2) (2012) 1094–1121.

[32] D. Needell, R. Vershynin, Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit, Foundations of computational mathematics 9 (3) (2009) 317–334.

[33] H. Rabah, A. Amira, A. Ahmad, Design and implementaiton of a fall detection system using compressive sensing and shimmer technology, in: Microelectronics (ICM), 2012 24th International Conference on, IEEE, 2012, pp. 1–4.

[34] K. P. Murphy, Machine learning: a probabilistic perspective, MIT press, 2012.

[35] D. Coppersmith, S. J. Hong, J. R. Hosking, Partitioning nominal attributes in decision trees, Data Mining and Knowledge Discovery 3 (2) (1999) 197–217.

[36] A. Majumdar, R. K. Ward, Energy efficient eeg sensing and transmission for wireless body area networks: A blind compressed sensing approach, Biomedical Signal Processing and Control 20 (2015) 1–9.

[37] M. Balouchestani, S. Krishnan, Robust compressive sensing algorithm for wireless surface electromyography applications, Biomedical Signal Processing and Control 20 (2015) 100–106.

[38] A. Adamo, G. Grossi, R. Lanzarotti, J. Lin, Ecg compression retaining the best natural basis k-coefficients via sparse decomposition, Biomedical Signal Processing and Control 15 (2015) 11–17.

[39] E. J. Candès, M. B. Wakin, An introduction to compressive sampling, Signal Processing Magazine, IEEE 25 (2) (2008) 21–30.

[40] A. Ramakrishnan, J. Satyanarayana, Reconstruction of eeg from limited channel acquisition using estimated signal correlation, Biomedical Signal Processing and Control 27 (2016) 164–173.

[41] T. Blumensath, M. E. Davies, Stagewise weak gradient pursuits, Signal Processing, IEEE Transactions on 57 (11) (2009) 4333–4346.

[42] T. Fushiki, Estimation of prediction error by using k-fold cross-validation, Statistics and Computing 21 (2) (2011) 137–146.

[43] G. Z. Karabulut, A. Yongacoglu, Sparse channel estimation using orthogonal matching pursuit algorithm, in: Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th, Vol. 6, IEEE, 2004, pp. 3880–3884.